NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS ACTIVE FIRES ARP SOFTWARE (D36981 Rev A)

CDRL No. A032

Northrop Grumman Space & Mission Systems Corporation
One Space Park
Redondo Beach, California 90278

This documentation/technical data was developed pursuant to Contract Number F04701-02-C-0502 with the US Government. The US Government's rights in and to this copyrighted data are as specified in DFAR 252.227-7013, which was made part of the above contract.

This document has been identified per the NPOESS Common Data Format Control Book – External Volume 5 Metadata, D34862-05, Appendix B as a document to be provided to the NOAA Comprehensive Large Array-data Stewardship System (CLASS) via the delivery of NPOESS Document Release Packages to CLASS.

The information provided herein does not contain technical data as defined in the International Traffic in Arms Regulations (ITAR) 22 CFR 120.10.

This document has been approved by the Unites States Government for public release in accordance with NOAA NPOESS Integrated Program Office.

Distribution: Statement A: Approved for public release; distribution is unlimited.

Northrop Grumman
Raytheon
DOC • DoD
NASA

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR VIIRS ACTIVE FIRES ARP SOFTWARE (D36981 Rev A)

PREPARED BY:
Dr Alain Sei, AM&S FV IP Lead
Paul D. Siebels, IDPS PRO SW Manager

ELECTRONIC APPROVAL SIGNATURES:

____
Roy Tsugawa   Date
Algorithm & Data Processing IPT Lead &
Algorithm Change Control Board Chairperson

____
Gerald J. Mulvey   Date
Senior Systems Engineer

NPOESS
Northrop Grumman
Raytheon
DOC • DoD
NASA

Revision/Change Record

Document Number D36981

Revision	Document Date	Revision/Change Description	Pages Affected
---	9-30-03	Initial Release.	All
A1	12-10-03	Changes to algorithm as a result of integrating into IPO model for IDPS.	All
A2	6-16-05	Reflects Science To Operational Code Conversion.	All
A3	9-25-06	Corrected Document ID number from D36981-IDP-001 to D36981 as directed by NGST DM.	All
A4	10-30-06	Updated grammar, etc for EDP Code PR.	All
A5	6-15-07	Logo, cleanup updates. Delivered to NGST.	All
A6	9-15-08	Reformatted to conform to template D41851. Updated graceful degradation and data quality monitoring sections. New cover sheet, update references, acronym list, prepare for peer review. Delivered to NGST.	All
A7	10-1-08	Added terrain corrected text to Geolocation input in Table 3. Prepared for TIM/ACCB.	All
A	10-29-08	ECR A-173. Incorporated interim changes and addressed TIM/ACCB comments. Approved for Public Release per Contracts letter 090618-01.	All

Table Of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

1.1 Objective

The purpose of the Operational Algorithm Description (OAD) document is to express, in computer-science terms, the remote sensing algorithms that produce the National Polar-Orbiting Operational Environmental Satellite System (NPOESS) end-user data products. These products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs). In addition, any Intermediate Products (IPs) produced in the process are also described in the OAD.

The science basis of an algorithm is described in a corresponding Algorithm Theoretical Basis Document (ATBD). The OAD provides a software description of that science as implemented in the operational ground system --- the Data Processing Element (DPE).

The purpose of an OAD is two-fold:

1. Provide initial implementation design guidance to the operational software developer
2. Capture the “as-built” operational implementation of the algorithm reflecting any changes needed to meet operational performance/design requirements

An individual OAD document describes one or more algorithms used in the production of one or more data products. There is a general, but not strict, one-to-one correspondence between OAD and ATBD documents.

1.2 Scope

The scope of this document is limited to describing the operational algorithm implementation required to create the VIIRS Active Fires Application Related Product (ARP). It provides a general overview and is intended to supplement in-line software documentation and interface control documentation for maintenance of the operational software.

The scope of this document is limited to the description of the core operational algorithm(s) required to create the VIIRS Active Fires ARP. The theoretical basis for this algorithm is described in Section 3 of VIIRS Active Fires: Fire Mask Algorithm Theoretical Basis Document (ATBD), D43770.

1.3 References

1.3.1 Document References

The science and system engineering documents relevant to the algorithms described in this OAD are listed in Table 1.

Table 1. Reference Documents

Document Title	Document Number/Revision	Revision Date
VIIRS Active Fires: Fire Mask Algorithm Theoretical Basis Document (ATBD) (ref P1187-TR-I-001)	D43770 Rev. A	18 Dec 2007
VIIRS Active Fires: Fire Mask Software Documentation	P1187-SW-I-001 Ver. 1.0.2	22 Sep 2003
VIIRS Algorithm Verification Status Report	D36812 Rev. 2.04	2 Dec 2003
Active Fires EDR Visible Infrared Imager/Radiometer Suite Science Grade Software Unit Test Document	D36980 Rev. ---	30 Sep 2003
VIIRS Radiometric Calibration Component Detailed Design Document	Y2490 Ver. 5 Rev. 4	30 Sep 2004

Document Title	Document Number/Revision	Revision Date
NPOESS Calibration/Validation Plan	D34484 Draft Version 3.0	17 Dec 2002
NPOESS EDR Performance Report	NPOESS.02.520.010 Ver. 3.3	2 Feb 2002
NPP EDR Production Report	D37005 Rev. C	16 Mar 2007
EDR Interdependency Report	D36385 Rev. C	7 Nov 2007
NPP Mission Data Format Control Book (MDFCB)	GSFC 429-05-02-42 R1	14 Apr 2006
CDFCB-X Volume I - Overview	D34862-01 Rev. B	27 Aug 2007
CDFCB-X Volume II – RDR Formats	D34862-02 Rev. B	27 Aug 2007
CDFCB-X Volume III – SDR/TDR Formats	D34862-03 Rev. A	27 Aug 2007
CDFCB-X Volume IV Part 1 – IP/ARP/GEO Formats	D34862-04-01 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 2 – Atmospheric, Clouds, and Imagery EDRs	D34862-04-02 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 3 – Land and Ocean/Water EDRs	D34862-04-03 Rev. A	10 Sep 2007
CDFCB-X Volume IV Part 4 – Earth Radiation Budget EDRs	D34862-04-04 Rev. A	10 Sep 2007
CDFCB-X Volume V - Metadata	D34862-05 Rev. B	27 Aug 2007
CDFCB-X Volume VI – Ancillary Data, Auxiliary Data, Reports, and Messages	D34862-06 Rev. C	10 Sep 2007
CDFCB-X Volume VII – Application Packets	D34862-07 Rev. ---	10 Sep 2007
NPOESS EDR Synergisms and Fusion Summary	D34837	20 Feb 2002
NPOESS IDP Segment Central Specification	SY10-0003 Rev. N	21 Sep 2007
NPOESS Modeling and Simulation Plan	D34475 Ver. 1.0	15 Mar 2002
NPOESS Scene Generation Development Report	D34861 Ver. 1.0	11 Feb 2002
NPOESS Subcontract Management Plan	D34845 Ver. 1.0	18 Feb 2002
NPOESS System Specification	SY15-0007 Ver. M	18 Oct 2007
NPOESS System Test Plan	D31406 Rev. C	6 May 2005
NPP Command and Telemetry (C&T) Handbook	568423 Rev. A	5 Apr 2005
Data Processor Inter-subsystem Interface Control Document (DPIS ICD)	D35850 Rev. U.2	27 Aug 2008
D35836_G_NPOESS_Glossary	D35836_G Rev. G	10 Sep 2008
D35838_G_NPOESS_Acronyms	D35838_G Rev. G	10 Sep 2008
NGST/SE technical memo – VCM_vs_VFM_Memo	NP-EMD.2004.510.0061	20 Dec 2004

1.3.2 Source Code References

The science and operational code and associated documentation relevant to the algorithms described in this OAD are listed in Table 2.

Table 2. Source Code References

Reference Title	Reference Tag/Revision	Revision Date
VIIRS Fire Mask Unit Test Data	New	30 Sep 2003
ClearCase Configuration Controlled Source Code	Version 1.0.2	23 Sep 2003
VIIRS Fire Mask science-grade software (initial drop)	NGST_2_3	30 Sep 2003
VIIRS Fire Mask operational software	B1.5	June 2007

2.0 ALGORITHM OVERVIEW

This document provides the operational algorithm description for the Fire Mask Unit of the VIIRS Active Fires algorithms, which produces the VIIRS Active Fires Application Related Product (ARP). The Fire Mask Algorithm, based upon the MODIS Fire Mask Version 4 code, produces a fire/ no fire mask from Visible and Infrared Imaging Radiometer Suite (VIIRS) data. The VIIRS Fire Mask (VFM) algorithm described here provides one of the three parameters identified in the NPOESS Active Fires ARP specification (40.6.4.1): the latitude and longitude of pixels containing active fires. The VIIRS Fire Mask is computed after the RDR and SDR processing is complete. The processing relationship is illustrated in Figure 1 below.

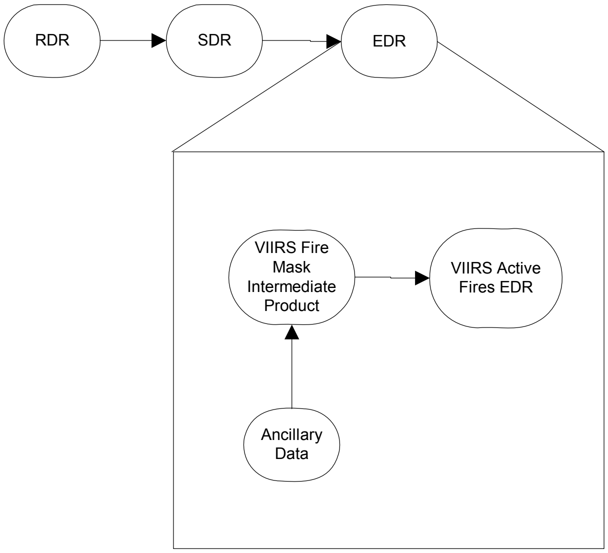

Figure 1. Processing Chain Associated with VIIRS Active Fires ARP

2.1 Fire Mask Algorithm Related Product Description

2.1.1 Interfaces

The Fire Mask algorithm is part of the Masks Controller process that is initiated by the Infrastructure (INF) Software Item (SI) to begin processing the data. The INF SI provides tasking information to the algorithm indicating which granule will be processed. The Data Management (DMS) SI provides data storage and retrieval capability. The interface to these SIs is implemented by a library of C++ classes. The interfaces to these other SIs are shown in Figure 2.

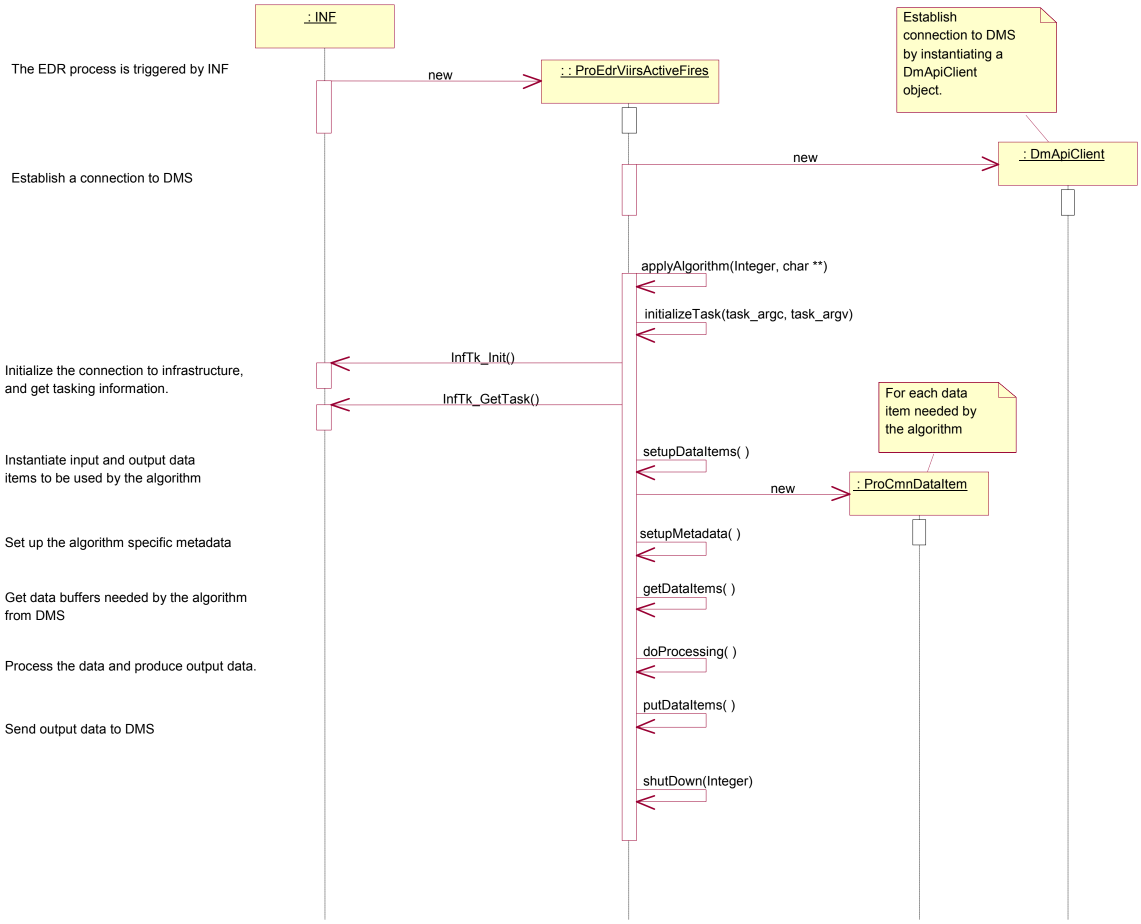
: INF
: : ProEdrViirsActiveFires
: DmApiClient
: ProCmnDataItem
Establish connection to DMS by instantiating a DmApiClient object.
For each data item needed by the algorithm
The EDR process is triggered by INF
new
Establish a connection to DMS
new
applyAlgorithm(Integer, char **)
initializeTask(task_argc, task_argv)
InfTk_Init()
Initialize the connection to infrastructure, and get tasking information.
InfTk_GetTask()
Instantiate input and output data items to be used by the algorithm
setupDataItems( )
new
Set up the algorithm specific metadata
setupMetadata( )
Get data buffers needed by the algorithm from DMS
getDataItems( )
Process the data and produce output data.
doProcessing( )
Send output data to DMS
putDataItems( )
shutDown(Integer)

Figure 2. IPO Model Interface to INF and DMS

2.1.1.1 Inputs

The Fire Mask algorithm requires several types of input data to compute the Fire Mask; they are summarized in Table 3 below. The input look-up tables required for Active Fires are described in Table 4. Format details for these data are documented in the Data Processor Inter-subsystem Interface Control Document (DPIS ICD), D35850.

A high-quality Land-Water Mask is required as ancillary data for VFM processing. The prototype algorithm uses the MODIS level 1B Land-Water Mask. Only a subset of the parameters is used: the land, water, and ephemeral water flags. This mask is not of sufficient accuracy to be used in the operational algorithm. The United States Geological Survey 30 arc-second terra-eco data item is used until the VIIRS Surface Type IP is available.

Table 3. VIIRS Fire Mask Inputs

Input	Type	Description	Units/Valid Range
VIIRS M5 SDR Reflectance	Float32 * [768 * 3200]	VIIRS Calibrated Top of the atmosphere (TOA) Reflectances	Unitless 0.0 – 1.6
VIIRS M7 SDR Reflectance	Float32 * [768 * 3200]	VIIRS Calibrated Top of the atmosphere (TOA) Reflectances	Unitless 0.0 – 1.6
VIIRS M11 SDR Reflectance	Float32 * [768 * 3200]	VIIRS Calibrated Top of the atmosphere (TOA) Reflectances	Unitless 0.0 – 1.6
VIIRS M13 SDR Brightness Temperature	Float32 * [768 * 3200]	VIIRS Top of the atmosphere (TOA) brightness temperatures	K 192.0 – 683.0
VIIRS M15 SDR Brightness Temperature	Float32 * [768 * 3200]	VIIRS Top of the atmosphere (TOA) brightness temperatures	K 192.0 – 683.0
VIIRS M16 SDR Brightness Temperature	Float32 * [768 * 3200]	VIIRS Top of the atmosphere (TOA) brightness temperatures	K 192.0 – 683.0
VIIRS Mod terrain corrected Geolocation Latitude	Float32 * [768 * 3200]	Latitude of each pixel (positive North)	Degree -90 - 90
VIIRS Mod terrain corrected Geolocation Longitude	Float32 * [768 * 3200]	Longitude of each pixel (positive East)	Degree -180 - 180
VIIRS Mod terrain corrected Geolocation satAzimuth	Float32 * [768 * 3200]	Azimuth angle (measured clockwise positive from North) to Satellite at each pixel position	Degree -180 - 180
VIIRS Mod terrain corrected Geolocation satZenith	Float32 * [768 * 3200]	Zenith angle to Satellite at each pixel position	Degree 0 - 180
VIIRS Mod terrain corrected Geolocation sunAzimuth	Float32 * [768 * 3200]	Azimuth angle of sun (measured clockwise positive from North) at each pixel position	Degree -180 - 180
VIIRS Mod terrain corrected Geolocation sunZenith	Float32 * [768 * 3200]	Zenith angle of sun at each pixel position	Degree 0 - 180
VIIRS Gip QstLwmGran Qstlwm	Uint8 * [768 * 3200]	Combined quarterly surface type	Unitless 1 - 20

Table 4. Input Look-Up Tables Required for Active Fires

Input	Type	Description	Original Source
Threshold Look-upTable	Data Table	LUT providing thresholds for the algorithm	Off-Line Processing

2.1.1.2 Outputs

2.1.1.2.1 VIIRS Active Fires ARP

The VIIRS Active Fires ARP produces output fields described in Table 5. The output is in the form of a vector for each pixel that Fire is detected.

Table 5. VIIRS Active Fires ARP Output File Content

Output	Description
List of pixels	List of values for each pixel with fire detected. The value is defined in Table 6

Table 6 details the VIIRS Active Fires ARP fields and descriptions.

Table 6. VIIRS Active Fires ARP Fields and Descriptions

Byte	Bit	Flag Description Key	Result
0-3	0-31	Latitude of Fire Pixel	Latitude(32-bit floating point)
4-7	32-63	Longitude of Fire Pixel	Longitude(32-bit floating point)
8-11	64-95	Index number of the row this fire pixel originated from	Row(32-bit signed integer)
12-15	96-127	Index number of the column this fire pixel originated from	Column(32-bit signed integer)
16	0	Adjacent Cloud Flag	0 = No 1 = Yes
	1	Adjacent Water Flag	0 = No 1 = Yes
	2-5	Search Window Size	Binary value between 1-10
	6	Sun Glint	0 = No 1 = Yes
	7	Sun Glint Override	0 = No 1 = Yes
17	0	Fire Test 1 Valid	0 = No 1 = Yes
	1	Fire Test 2 Valid	0 = No 1 = Yes
	2	Fire Test 3 Valid	0 = No 1 = Yes
	3	Fire Test 4 Valid	0 = No 1 = Yes
	4	Fire Test 5 Valid	0 = No 1 = Yes
	5	Fire Test 6 Valid	0 = No 1 = Yes
	6	Input Data Quality	0 = Good 1 = Poor
	7	Day/Night	0 = Night 1 = Day
18	0	False Alarm Override	0 = No 1 = Yes
	1	Water Contamination Override	0 = No 1 = Yes
	2-7	Spare Bits	Initialized to 0
19	0-7	Fire Detection Confidence	0 – 100%

2.1.1.2.2 VIIRS Fire Mask IP (Heap Data Item)

The VIIRS Fire Mask produces the Fire Mask IP output as a heap data item which is internal to the algorithm. The fields are described in Table 7. The enumerated outputs in the Fire Mask are CLASS_NOPROC_MISS, CLASS_NOPROC_SCAN, CLASS_NOPROC_OTHER, CLASS_WATER, CLASS_CLOUD, CLASS_NO_FIRE, CLASS_UNKNOWN, CLASS_FIRE_LOW, CLASS_FIRE_MED, and CLASS_FIRE_HIGH. The quality assurance bitmap records information regarding the success/failure of the various fire tests; its structure is described in Table 8.

Table 7. VIIRS Fire Mask IP Heap Output Content

Output	Description
Fire Mask	Enumerated values representing the fire classification of each pixel in the granule
qcFlags	Quality Assurance bitmap with test information for each pixel in the granule: Missing – 0 Scan – 1 Other – 2 Water – 3 Cloud – 4 No Fire – 5 Unknown – 6 Fire Low – 7 Fire Medium – 8 Fire High - 9

Table 8. VFM IP Quality Flag Bits and Descriptions

Byte	Bit	Flag Description Key	Result
0	0	Adjacent Cloud Flag	0 = No 1 = Yes
	1	Adjacent Water Flag	0 = No 1 = Yes
	2-5	Search Window Size	Binary value between 1-10
	6	Sun Glint	0 = No 1 = Yes
	7	Sun Glint Override	0 = No 1 = Yes
1	0	Fire Test 1 Valid	0 = No 1 = Yes
	1	Fire Test 2 Valid	0 = No 1 = Yes
	2	Fire Test 3 Valid	0 = No 1 = Yes
	3	Fire Test 4 Valid	0 = No 1 = Yes
	4	Fire Test 5 Valid	0 = No 1 = Yes
	5	Fire Test 6 Valid	0 = No 1 = Yes
	6	Input Data Quality	0 = Good 1 = Poor
	7	Day/Night	0 = Night 1 = Day
2	0	False Alarm Override	0 = No 1 = Yes

Byte	Bit	Flag Description Key	Result
	1	Water Contamination Override	0 = No 1 = Yes
	2-7	Spare Bits	Initialized to 0
3	0-7	Fire Detection Confidence	0 – 100%

2.1.2 Algorithm Processing

Inputs to the algorithm are measured VIIRS normalized reflectances, brightness temperatures, and ancillary data as well as algorithm threshold information.

2.1.2.1 Main Module - ProEdrViirsActiveFires.cpp

This is the derived algorithm for the Active Fires algorithm and is a subclass of the ProCmnAlgorithm class. The class creates a list of input data items that are read from DMS and passes all of the required data into the algorithm itself. When the algorithm has finished processing the data, an output item is written to DMS that contains the fire mask.

2.1.2.2 ProEdrViirsActiveFires::detect(…)

This is the main fire detection algorithm that wraps a modified version of MODIS Fire Mask code version 4. Initialize the inext and iprev arrays for each scan. Examine each pixel in the granule. If the pixel is in a bow-tie deletion region, or if either T13B or T15 data is missing, or if the pixel is classified water, or if pixel is cloudy, then skip to the next pixel in the scan direction. Non-water pixels containing valid data are counted as land. If the pixel was not classified as a potential fire, then skip to the next pixel. Choose the day or night thresholds depending on the day or night flag.

Next, compute the background statistics for all potential fire pixels. During the day compute the NDVI for all of the background pixels with R7 > 0. Use the NDVI to determine if any of these pixels are water pixels. If there are any background fires, compute statistics for the background fire pixels. If there are any valid background pixels, calculate the means MeanT13B, MeanT15, and MeanDT, and mean absolute deviations MadT13B, MadT15, and MadDT, for T13B, T15, and DT = T13B – T15. De-mean the T13B data and normalize it by the mean absolute deviation giving

$$z21 = (T13- MeanT13B)/(MadT13B - eps).$$

These values are used to compute the contextual fire tests for the pixel:

$test2 = DT > (MeanDT + test2\_sigma * MadDT),$
$test3 = DT > (MeanDT + minbkg\_DT),$
$test4 = T13B > (MeanT13B + test4\_sigma * MadT13B),$
$test5 = T15 > (MeanT15 + MadT15 - devrp\_T15),$ and
$test6 = BkgFireMadT13B > test6\_sigma.$

For the pixel compute the absolute fire test given by

$$test1=T13>thresholdT13.$$

The values for minbkg_DT, devrp_T15, and thresholdT13 are different for day and night.
If there are valid background pixels then use both absolute and contextual fire tests. For these tests, a fire is detected if test1 is true or if test2, test3, and test4 are all true and test5 or test6 is true. If there are no valid background pixels, use only the absolute fire test; the pixel is a fire

pixel if test1 is true. If no fire is detected and there are valid background pixels, then set the fire mask to “no fire” otherwise set the fire mask to “unknown”.

Test the declared fire pixels for false alarms. If the sun glint is large or if it is moderate glint and the pixel is near water pixels, then reject the pixel and set the fire mask to “no fire”. Override detection if it is likely that this is a false alarm caused by excessive rejection of legitimate background pixels. During the day, if the absolute fire test is not true, the fraction of valid background pixels, given by nvalid / nvalid_land, is less than the threshold bkgoverride_fvalid, the number of background fires is greater than the threshold bkgoverride_nbfire, BkgFireMeanT13B is less than the threshold bkgoverride_MeanT13B, BkgFireMadT13B is less than the threshold bkgoverride_MadT13B, the band 7 normalized reflectances greater than bkgoverride_R7, and T13B is less than (BkgFireMeanT13B + bkgoverride_sigmaT13B * BkgFireMadT13B) then the fire is rejected and the fire mask for the pixel is set to “no fire”. Also during the day, if the absolute fire test is not true and there are water pixels in the background, then reject the fire as a false alarm and set the fire mask for the pixel to “no fire”. If a false alarm is detected, skip to the next pixel. Compute the fire confidence and use it to set the fire mask confidence level.

2.1.2.3 ProEdrViirsActiveFires::prescreen(…)

This is a wrapper around modified MODIS Fire Mask Version 4 code designed to detect potential fire pixels. The routine loops over each pixel in the granule. If the solar zenith angle is < 85°, then it is day and check for sun glint. Calculate the relative azimuth angle between the sun and satellite directions and calculate the glint level by invoking the method sunGlint(…). The land-water mask is checked to see if the pixel is over land. For the MOD03 land mask, treat "land" and "ephemeral water" as land. If the pixel is water, then skip to the next pixel. Next check if the pixel is cloudy. Use the internal cloud test if T16 is greater than zero by invoking the method iscloud(R5, R7, T16). If the internal tests mark the pixel as cloudy, skip to the next pixel. Different thresholds are used for day or night. The difference between the M13 and M15 brightness temperatures are given by DT=T13-T15. If T13 > T13threshold, DT > Dtthreshold, and Test3 = R7 < R7threshold, then mark the pixel as a potential fire. Next perform the background pixel tests. If T13 > T13backgroundThreshold and DT > DtbackgroundThreshold, then mark the pixel as a potential background fire otherwise mark it as a valid background pixel.

2.1.2.4 ProEdrViirsActiveFires::bkgPixels(…)

This is a wrapper around modified MODIS Fire Mask Version 4 code designed to estimate background by searching the neighboring area. The search for valid background pixels is made by placing a window around the possible fire pixel. The initial window size starts as a 3 x 3 ring. The number of valid background pixels in the window, nvalid, is then retrieved by a call to the getValidBkgPixels(…) method. The minimum number of background pixels required in the search window is given by

$$\text{minvalid} = \text{Valid\_Win\_Ratio}\ (\text{winsize}^{2} - \text{Win\_Exclude\_Size})$$

where Valid_Win_Ratio is the minimum fraction of valid pixels in the window and Win_Exclude_Size is the number of excluded pixels in the window. If minvalid is less than the minimum number of valid pixels, Valid_Win_Size, then set it equal to Valid_Win_Size. If nvalid is greater than minvalid, then return nvalid, otherwise increase the size of the window by 2 pixels on a side and try again. If the window is larger than the maximum window size then return nvalid = 0.

2.1.2.5 ProEdrViirsActiveFires::getValidBkgPixels(…)

This is a wrapper around modified MODIS Fire Mask Version 4 code designed to get background by searching the neighboring area. Since the background search starts with a 3 x 3 window or r = 1, as it is called with larger values of r, only the outer edge has to be searched each time. The routine was modified to consider bow-tie deletion areas and cross-scan interleave. If the background search window extends past either end of the scan line then set it equal to the scan line. The possible fire pixel is located at (ic,jc). The index of the scan for the current row, ic, is calculated. For the top edge of the window, set the cross-scan index to i = ic – r, where r is the “radius” of the window. If i < top[scan][jc] then look in the previous scan, handling bow-tie deletion. Examine pixels in the scan direction with $jc – r \leq j \leq jc + r$. The cross-scan index in the previous scan is given by

i += 1 - top[scan][jc] + prev[scan][j]

where prev[scan][j] is the nearest neighbor pixel in the previous scan. If pixel (i,j) is a valid background pixel then retrieve T13B and T15 from the data structure and store them in the appropriate buffers. Also store T13B – T15 in BuffDT and increment nvalid. If pixel (i,j) has R7 > 0 then retrieve R5, R7, and R11 from the data structure, store them in the appropriate buffers, and increment nref. If pixel (i,j) is a background fire pixel then retrieve T13B and T15 from the data structure and store them in the appropriate buffers. Also store T13B – T15 in BkgFireBuffDT and increment nbfire. If pixel (i,j) is a water pixel then increment nwater. For the bottom edge of the window, set the cross scan index to i = ic + r. If i > bottom[scan][jc] then look in the next scan, handling bow-tie deletion. Again examine pixels in the scan direction with $jc – r \leq j \leq jc + r$. The cross-scan index in the next scan is given by

i = ic + r - Bottom[scan][jc] - 1 + next[scan][j]

where next[scan][j] is the nearest neighbor pixel in the next scan. Store the brightness temperatures and normalized reflectances as above, and increment the appropriate counters. If doing 3 x 3 window, i.e. r = 1, skip pixels adjacent to the candidate fire pixel in the along-scan direction. The left edge of the window is given by jl = jc – r, while the right edge is given by jr = jc + r. For the left side, search in cross-scan with $ic - r + 1 \leq i < ic + r$. The value of the row index, icorr, is computed to look in the appropriate previous, next or current scan. For pixel (icorr,jl) store the brightness temperatures and normalized reflectances as was done for the top edge, and increment the appropriate counters. Treat the right side as the left, and retrieve the data for pixel (icorr,jr) and add it to the buffers, incrementing the appropriate counters.

2.1.2.6 ProEdrViirsActiveFires::countAdjacentPixels(…)

This method is a Wrapper around modified MODIS code. It counts the number of cloud and water pixels in the eight adjacent pixels to the target pixel given by (ic, jc). If it is the bottom or top pixel in the cross-scan direction, including the effects of bow-tie deletion, the nearest neighbor pixels are taken from the previous or next scan. The water and cloud flags of each adjacent pixel are examined and the total number of pixels marked as water is returned.

2.1.2.7 ProEdrViirsActiveFires::computeStats(…)

This is the method to compute mean and mean absolute deviation of input data in an arbitrary input array x containing n values. The mean is given by

$$\text{mean} = \frac{1}{n}\sum_{i=0}^{n-1} x_i \ .$$

The mean absolute deviation of array x is given by

$$\mathrm{mad} = \frac{1}{n}\sum_{i=0}^{n-1}\left|x_i - \mathrm{mean}\right|.$$

2.1.2.8 ProEdrViirsActiveFires::sunGlint(…)

This method is a wrapper around original MODIS code to test for sun glint. The glint angle is given by

cos(GlintAng) = cos(view zenith angle) cos(solar zenith angle) - sin(view zenith angle) sin(solar zenith angle) cos(relative azimuth angle).

For very small glint angles automatically assume that there is glint. The level of the glint is set by tests based on the glint angle and the normalized reflectances R5, R7, and R11. The program returns the glint level.

2.1.2.9 ProEdrViirsActiveFires::fixang(…)

This is the method for converting angles from --2π ≤ ang ≤ 2π to -π ≤ ang ≤ π. If ang > π, then return ang - 2π or if ang ≤ -π then return ang + 2π otherwise, return the input value of ang.

2.1.2.10 ProEdrViirsActiveFires::confidence(…)

This method is a wrapper around the MODIS confidence test routine. It produces fire detection confidence levels. First, the method calculates the normalized, de-meaned difference in T13 – T15, zDT. This value is given by

$$\mathrm{zDT} = \frac{\mathrm{DT} - \mathrm{MeanDT}}{\mathrm{MadDT} + \mathrm{eps}}.$$

Next the S ratio function is used to calculate two individual confidence tests given by C2 = S(z21, 3, 6) and C3 = S(zDT, 3.5, 6). During the day, calculate three more individual confidence tests given by

C1 = S(T13B, 310, 340),
C4 = 1 - S(AdjCloud, 0, 6), and
C5 = 1 - S(AdjWater, 0, 6).

The confidence is then given by taking the geometric mean of the results of the five tests:

$$C = \sqrt[5]{C1 \cdot C2 \cdot C3 \cdot C4 \cdot C5}.$$

At night, calculate another individual confidence test given by C1 = S(T13B, 305, 320). The confidence is then given by taking the geometric mean of the results of the three tests:

$$C = \sqrt[3]{C1 \cdot C2 \cdot C3}.$$

The confidence results are then converted to percent and returned by the routine.

2.1.2.11 ProEdrViirsActiveFires::S(…)

This is the method to compute the ratio S. The procedure is derived from original MODIS code. If $x \leq a$, then return 0; if $x \geq b$ return 1; otherwise, return $(x - a) / (b - a)$.

2.1.2.12 ProEdrViirsActiveFires::computeNDVI(…)

This is the method derived from original MODIS code to compute the NDVI ratio for n pixels. The routine calculates the NDVI for each of the n pixels where the NDVI is given by

$$NDVI = (R7 - R5) / (R7 + R5).$$

2.1.2.13 ProEdrViirsActiveFires::bkgWaterPix(…)

This is the method derived from original MODIS code to determine if water is present in a region. Given R7, R11, and NDVI then a pixel is considered water if R7 < bkgwater_R7 and R11 < bkgwater_R11 and NDVI < bkgwater_NDVI. For n input pixel values the total number of pixels, nwater, that pass the test is returned. The values of bkgwater_R7, bkgwater_R11, and bkgwater_NDVI are read from the thresholds data item.

2.1.2.14 ProEdrViirsActiveFires::iscloud(…)

This is the method to test a pixel for presence of clouds. There are four tests performed using R5, R7, and T16. If (R5 + R7) > iscloud_test1, or T16 < iscloud_test2, or (R5 + R7) > iscloud_test3 and T16 < iscloud_test4 then the pixel is declared to be cloudy. The values of iscloud_test1, iscloud_test2, iscloud_test3, and iscloud_test4 are set during initialization.

2.1.3 Graceful Degradation

2.1.3.1 Graceful Degradation Inputs

There is one case where input graceful degradation is indicated in the Active Fires Mask.

1. An input retrieved for the algorithm had its N_Graceful_Degradation metadata field set to YES (propagation).

Table 9. Graceful Degradation

Input Data Description	Baseline Data Source	Primary Backup Data Source	Secondary Backup Data Source	Tertiary Backup Data Source	Graceful Degradation Done Upstream
Quarterly Surface Type IP	VIIRS_GD_08.4.3 VIIRS	N/A	N/A	N/A	N/A
Land-Water Mask	VIIRS_GD_27.4.1 MODIS	N/A	N/A	N/A	N/A

2.1.3.2 Graceful Degradation Processing

None.

2.1.3.3 Graceful Degradation Outputs

None.

2.1.4 Exception Handling

Any exceptions or errors are reported to IDPS using the appropriate INF API. All input is assumed to be available until the graceful degradation plan has been implemented.

- The software checks for bad or missing pixels (denoted by a fill value) and missing ancillary data. It does not perform a retrieval in such cases.

2.1.5 Data Quality Monitoring

Each algorithm uses specific criteria contained in a Data Quality Threshold Table (DQTT) to determine when a Data Quality Notification (DQN) is produced. The DQTT contains the threshold used to trigger the DQN as well as the text contained in the DQN. If a threshold is met, the algorithm stores a DQN in DMS indicating the test(s) that failed and the value of the DQN attribute. For more algorithm specific detail refer to the CDFCB-X.

2.1.6 Computational Precision Requirements

The current implementation of the Active Fire algorithm performs computations using 32-bit floating-point precision values. The SDR geolocation information is passed through and no change of precision occurs in the science code.

2.1.7 Algorithm Support Considerations

No specific software environment considerations are required.

2.1.8 Assumptions and Limitations

2.1.8.1 Assumptions

Overlapping pixels deleted from the down linked data stream are assumed to be replaced with fill values during SDR processing.

2.1.8.2 Limitations

None identified at this time.

3.0 GLOSSARY/ACRONYM LIST

3.1 Glossary

The current glossary for the NPOESS program, D35836_G_NPOESS_Glossary, can be found on eRooms. Table 10 contains those terms most applicable for this OAD.

Table 10. Glossary

Term	Description
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On NPOESS, an algorithm consists of: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code)
Algorithm Configuration Control Board (ACCB)	Interdisciplinary team of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Algorithm Implementation Process Lead, members include representatives from IWPTB, Systems Engineering & Integration IPT, System Test IPT, and IDPS IPT.
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the IWPTB facility. Delivered code is executed on compatible IWPTB computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.). *[Supplementary Definition]* An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more NPOESS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications. [Ref.: DoDD 5000.59-DoD Modeling and Simulation Management]
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by IWPTB, is developed into operational-grade code by the IDPS IPT.
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code.

Term	Description
Raw Data Record (RDR)	*[IORD Definition]* Full resolution digital sensor data, time referenced and earth located, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression. *[Supplementary Definition]* A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of NPOESS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/NPOESS data product (TDR, SDR or EDR). The former is described in an ATBD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor.
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure.
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.

Term	Description
Temperature Data Record (TDR)	*[IORD Definition]* Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts. *[Supplementary Definition]* A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.

3.2 Acronyms

The current acronym list for the NPOESS program, D35838_G_NPOESS_Acronyms, can be found on eRooms. Table 11 contains those terms most applicable for this OAD.

Table 11. Acronyms

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
ARP	Application Related Product
CDFCB-X	Common Data Format Control Book - External
DMS	Data Management Subsystem
DPIS ICD	Data Processor Inter-subsystem Interface Control Document
DQTT	Data Quality Test Table
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
LUT	Look-Up Table
MDFCB	Mission Data Format Control Book
QF	Quality Flag
SDR	Sensor Data Record
SI	International System of Units
TBD	To Be Determined
TBR	To Be Resolved
TOA	Top of the Atmosphere
VCM	VIIRS Cloud Mask
VFM	VIIRS Fire Mask

4.0 OPEN ISSUES

Table 12. TBXs

TBX ID	Title/Description	Resolution Date
None		